

REMARKS

Examiner Coulter telephoned on 28 January 2002, and indicated that a substitute specification was required in order to comply with 37 CFR 1.52 (e) for computer program listings exceeding 300 lines.

Applicant submits such substitute specification, along with a computer program listing appendix on CD in compliance with the rules.

Applicant also updates the reference to related applications in the first paragraph of the specification.

Accordingly, included with this paper are the following items:

- (1) SUBSTITUTE SPECIFICATION (without drawings);
- (2) TRANSMITTAL LETTER FOR ELECTRONIC DOCUMENT UNDER 37 CFR 1.52 (e); and
- (3) COPY 1 and COPY 2 of a CD-ROM labeled CMRC1006-2, including the computer program listing appendix.

The computer program listing appendix includes listings which are incorporated by reference into the specification. The listings are identical with those filed in the original application.

CONCLUSION

It is submitted that the present application is now in form for allowance, and such action is respectfully requested.

Respectfully submitted,

Date: February 22, 2002



Mark A. Haynes
Registration No. 30,846

HAYNES BEFFEL & WOLFELD LLP
P.O. Box 366
Half Moon Bay, CA 94019
(650) 712-0340 - telephone
(650) 712-0263 - facsimile

5

10

15

20

25

30

follows:

[0064] An example business interface definition BID based on a market participant document which conforms to an XML format is provided below.

5 The market participant DTD groups business information about market participants, associating contact and address information with a description of services and financial information. This business information is composed of names, codes, addresses, a dedicated taxonomic mechanism for describing business organization, and a pointer to terms of business. In addition, the
10 services identified by the market participant DTD will specify the input and output documents which that participant is expected respond to and produce. Thus, documents which define schema using an exemplary common business language for a market participant DTD, a service DTD, and a transaction document DTD specified in XML with explanatory comments are set forth as
15 LISTING 1, in the file named LISTING COMBINED.txt in the accompanying computer program listing appendix.[follow:

Market Participant Sample

20 <!DOCTYPE SCHEMA SYSTEM "bidl.dtd">
<SCHEMA>
<H1>Market Participant Sample BID</H1>
<META
WHO.OWNS="Veo Systems" WHO.COPYRIGHT="Veo Systems"
25 WHEN.COPYRIGHT="1998" DESCRIPTION="Sample BID"
WHO.CREATED="*" WHEN.CREATED="*"
WHAT.VERSION="*" WHO.MODIFIED="*"
WHEN.MODIFIED="*" WHEN.EFFECTIVE="*"
WHEN.EXPIRES="*" WHO.EFFECTIVE="*"
30 WHO.EXPIRES="*">
</META>

<PROLOG>
<XMLDECL STANDALONE="no"></XMLDECL>
<DOCTYPE NAME="market.participant">
35 <SYSTEM>markpart.dtd</SYSTEM></DOCTYPE>
</PROLOG>

<DTD NAME="markpart.dtd">
<H2>Market Participant</H2>
40 <H3>Market Participant</H3>

<ELEMENTTYPE NAME="market.participant">
 <EXPLAIN><TITLE>A Market Participant</TITLE>
 <SYNOPSIS>A business or person and its service interfaces.</SYNOPSIS>
 5 <P>A market participant is a document definition that is created to describe a business and at
 least one person with an email address, and it presents a set of pointers to service interfaces
 located on the network. In this example, the pointers have been resolved and the complete BID
 is presented here.</P></EXPLAIN>
 <MODEL><CHOICE>
 10 <ELEMENT NAME="business"></ELEMENT>
 <ELEMENT NAME="person"></ELEMENT>
 </CHOICE></MODEL></ELEMENTTYPE>

<H3>Party Prototype</H3>
 <PROTOTYPE NAME="party">
 15 <EXPLAIN><TITLE>The Party Prototype</TITLE></EXPLAIN>
 <MODEL><SEQUENCE>
 <ELEMENT NAME="party.name" OCCURS="+"></ELEMENT>
 <ELEMENT NAME="address.set"></ELEMENT>
 </SEQUENCE></MODEL>
 20 </PROTOTYPE>

<H3>Party Types</H3>
 <ELEMENTTYPE NAME="business">
 <EXPLAIN><TITLE>A Business</TITLE>
 25 <SYNOPSIS>A business (party) with a business number attribute.</SYNOPSIS>
 <P>This element inherits the content model of the party prototype and adds a business number
 attribute, which serves as a key for database lookup. The business number may be used as a
 cross-reference to/from customer id, credit limits, contacts lists, etc.</P></EXPLAIN>
 <EXTENDS HREF="party">
 30 <ATTDEF NAME="business.number"><REQUIRED></REQUIRED></ATTDEF>
 </EXTENDS>
 </ELEMENTTYPE>

<H3>Person Name</H3>
 35 <ELEMENTTYPE NAME="person">
 <EXPLAIN><TITLE>A Person</TITLE></EXPLAIN>
 <EXTENDS HREF="party">
 <ATTDEF NAME="SSN"><IMPLIED></IMPLIED></ATTDEF>
 </EXTENDS>
 40 </ELEMENTTYPE>

<H3>Party Name</H3>
 <ELEMENTTYPE NAME="party.name">
 <EXPLAIN><TITLE>A Party's Name</TITLE>
 45 <SYNOPSIS>A party's name in a string of character.</SYNOPSIS></EXPLAIN>
 <MODEL><STRING></STRING></MODEL>
 </ELEMENTTYPE>

<H3>Address Set</H3>
 50 <ELEMENTTYPE NAME="address.set">
 <MODEL><SEQUENCE>
 <ELEMENT NAME="address.physical"></ELEMENT>
 <ELEMENT NAME="telephone" OCCURS="*"></ELEMENT>
 <ELEMENT NAME="fax" OCCURS="*"></ELEMENT>

<ELEMENT NAME="email" OCCURS="*"></ELEMENT>
 <ELEMENT NAME="internet" OCCURS="*"></ELEMENT>
 </SEQUENCE></MODEL>
 </ELEMENTTYPE>

5

<H3>Physical Address</H3>
 <ELEMENTTYPE NAME="address.physical">
 <EXPLAIN><TITLE>Physical Address</TITLE>
 <SYNOPSIS>The street address, city, state, and zip code.</SYNOPSIS></EXPLAIN>
 <MODEL><SEQUENCE>
 <ELEMENT NAME="street"></ELEMENT>
 <ELEMENT NAME="city"></ELEMENT>
 <ELEMENT NAME="state"></ELEMENT>
 <ELEMENT NAME="postcode" OCCURS="?"></ELEMENT>
 <ELEMENT NAME="country"></ELEMENT>
 </SEQUENCE></MODEL>
 </ELEMENTTYPE>

10

<H3>Street</H3>
 <ELEMENTTYPE NAME="street">
 <EXPLAIN><TITLE>Street Address</TITLE>
 <SYNOPSIS>Street or postal address.</SYNOPSIS></EXPLAIN>
 <MODEL><STRING></STRING></MODEL>
 </ELEMENTTYPE>

15

<H3>City</H3>
 <ELEMENTTYPE NAME="city">
 <EXPLAIN><TITLE>City Name or Code</TITLE>
 <P>The city name or code is a string that contains sufficient information to identify a city within
 a designated state.</P>
 </EXPLAIN>
 <MODEL><STRING></STRING></MODEL>
 </ELEMENTTYPE>

20

<H3>State</H3>
 <ELEMENTTYPE NAME="state">
 <EXPLAIN><TITLE>State, Province or Prefecture Name or Code</TITLE>
 <P>The state name or code contains sufficient information to identify a state within a designated
 country.</P></EXPLAIN>
 <MODEL><STRING DATATYPE="COUNTRY.US.SUBENTITY"></STRING></MODEL>
 </ELEMENTTYPE>

25

<H3>Postal Code</H3>
 <ELEMENTTYPE NAME="postcode">
 <EXPLAIN><TITLE>Postal Code</TITLE>
 <P>A postal code is an alphanumeric code, designated by an appropriate postal authority, that is
 used to identify a location or region within the jurisdiction of that postal authority. Postal
 authorities include designated national postal authorities.</P></EXPLAIN>
 <MODEL><STRING DATATYPE="string"></STRING></MODEL>
 </ELEMENTTYPE>

30

<H3>Country</H3>
 <ELEMENTTYPE NAME="country">
 <EXPLAIN><TITLE>Country Code</TITLE>

35

40

45

50

<P>A country code is a two-letter code, designated by ISO, that is used to uniquely identify a country.</P></EXPLAIN>

<MODEL><STRING DATATYPE="country"></STRING></MODEL>
</ELEMENTTYPE>

<H3>Network Addresses</H3>

<ELEMENTTYPE NAME="telephone">

<EXPLAIN><TITLE>Telephone Number</TITLE>

<P>A telephone number is a string of alphanumerics and punctuation that uniquely identifies a telephone service terminal, including extension number.</P></EXPLAIN>

<MODEL><STRING></STRING></MODEL>
</ELEMENTTYPE>

<H3>Fax</H3>

<ELEMENTTYPE NAME="fax">

<EXPLAIN><TITLE>Fax Number</TITLE>

<P>A fax number is a string of alphanumerics and punctuation that uniquely identifies a fax service terminal.</P>

</EXPLAIN>

<MODEL><STRING></STRING></MODEL>
</ELEMENTTYPE>

<H3>Email</H3>

<ELEMENTTYPE NAME="email">

<EXPLAIN><TITLE>Email Address</TITLE>

<P>An email address is a datatype-constrained string that uniquely identifies a mailbox on a server.</P></EXPLAIN>

<MODEL><STRING DATATYPE="email"></STRING></MODEL>
</ELEMENTTYPE>

<H3>Internet Address</H3>

<ELEMENTTYPE NAME="internet">

<EXPLAIN><TITLE>Internet Address</TITLE>

<P>An Internet address is a datatype-constrained string that uniquely identifies a resource on the Internet by means of a URL.</P></EXPLAIN>

<MODEL><STRING DATATYPE="url"></STRING></MODEL>
</ELEMENTTYPE>

</DTD>

</SCHEMA>

Service Description Sample

<!DOCTYPE schema SYSTEM "bidl.dtd">

<SCHEMA>

<H1>Service Description Sample BID</H1>

<META

WHO.OWNS="Veo Systems"

WHO.COPYRIGHT="Veo Systems"

WHEN.COPYRIGHT="1998"

DESCRIPTION="Sample BID"

WHO.CREATED="*"

WHEN.CREATED="*"

WHAT.VERSION="*"

WHO.MODIFIED="*"

WHEN.MODIFIED="*"

WHEN.EFFECTIVE="*"

WHEN.EXPIRES="*"

WHO.EFFECTIVE="*"

WHO.EXPIRES="*"

```

</META>

<PROLOG>
<XMLDECL STANDALONE="no"></XMLDECL>
5 <DOCTYPE NAME="service">
<SYSTEM>service.dtd</SYSTEM></DOCTYPE>
</PROLOG>

10 <DTD NAME="service.dtd">
<H2>Services</H2>

<H3>Includes</H3>
<!-- INCLUDE><SYSTEM>comments.bim</SYSTEM></INCLUDE -->

15 <H3>Service Set</H3>
<ELEMENTTYPE NAME="service.set">
<EXPLAIN><TITLE>Service Set</TITLE>
<SYNOPSIS>A set of services.</SYNOPSIS></EXPLAIN>
<MODEL>
20 <ELEMENT NAME="service" OCCURS="+"></ELEMENT>
</MODEL></ELEMENTTYPE>

<H3>Services Prototype</H3>
<PROTOTYPE NAME="prototype.service">
25 <EXPLAIN><TITLE>Service</TITLE></EXPLAIN>
<MODEL><SEQUENCE>
<ELEMENT NAME="service.name"></ELEMENT>
<ELEMENT NAME="service.terms" OCCURS="+"></ELEMENT>
<ELEMENT NAME="service.location" OCCURS="+"></ELEMENT>
30 <ELEMENT NAME="service.operation" OCCURS="+"></ELEMENT>
</SEQUENCE></MODEL>
<!-- ATTGROUP><IMPLEMENTS
HREF="common.attrib"></IMPLEMENTS></ATTGROUP -->
</PROTOTYPE>

35 <H3>Service</H3>
<INTRO><P>A service is an addressable network resource that provides interfaces to specific
operations by way of input and output documents.</P></INTRO>
<ELEMENTTYPE NAME="service">
40 <EXPLAIN><TITLE>Service</TITLE>
<P>A service is defined in terms of its name, the location(s) at which the service is available,
and the operation(s) that the service performs.</P></EXPLAIN>
<MODEL><SEQUENCE>
<ELEMENT NAME="service.name"></ELEMENT>
45 <ELEMENT NAME="service.location"></ELEMENT>
<ELEMENT NAME="service.operation" OCCURS="+"></ELEMENT>
<ELEMENT NAME="service.terms"></ELEMENT>
</SEQUENCE></MODEL>
</ELEMENTTYPE>

50 <H3>Service Name</H3>
<ELEMENTTYPE NAME="service.name">
<EXPLAIN><TITLE>Service Name</TITLE>
<P>The service name is a human-readable string that ascribes a moniker for a service. It may be

```

employed is user interfaces and documentation, or for other purposes.</P></EXPLAIN>
<MODEL><STRING></STRING></MODEL>
</ELEMENTTYPE>

5 <H3>Service Location</H3>
<ELEMENTTYPE NAME="service.location">
<EXPLAIN><TITLE>Service Location</TITLE>
<SYNOPSIS>A URI of a service.</SYNOPSIS>
<P>A service location is a datatype-constrained string that locates a service on the Internet by
10 means of a URI.</P></EXPLAIN>
<MODEL><STRING DATATYPE="url"></STRING></MODEL>
</ELEMENTTYPE>

<H3>Service Operations</H3>
15 <INTRO><P>A service operation consists of a name, location and its interface, as identified by
the type of input document that the service operation accepts and by the type of document that it
will return as a result.</P></INTRO>
<ELEMENTTYPE NAME="service.operation">
<EXPLAIN><TITLE>Service Operations</TITLE>
20 <P>A service operation must have a name, a location, and at least one document type as an
input, with one or more possible document types returned as a result of the operation.</P>
</EXPLAIN>
<MODEL><SEQUENCE>
<ELEMENT NAME="service.operation.name"></ELEMENT>
25 <ELEMENT NAME="service.operation.location"></ELEMENT>
<ELEMENT NAME="service.operation.input"></ELEMENT>
<ELEMENT NAME="service.operation.output"></ELEMENT>
</SEQUENCE></MODEL>
</ELEMENTTYPE>

30 <H3>Service Operation Name</H3>
<ELEMENTTYPE NAME="service.operation.name">
<EXPLAIN><TITLE>Service Operation Name</TITLE>
<P>The service operation name is a human-readable string that ascribes a moniker to a service
35 operation. It may be employed in user interfaces and documentation, or for other
purposes.</P></EXPLAIN>
<MODEL><STRING></STRING></MODEL>
</ELEMENTTYPE>

40 <H3>Service Operation Location</H3>
<INTRO><P>The service location is a network resource. That is to say, a URI.</P></INTRO>
<ELEMENTTYPE NAME="service.operation.location">
<EXPLAIN><TITLE>Service Operation Location</TITLE>
<SYNOPSIS>A URI of a service operation.</SYNOPSIS>
45 <P>A service operation location is a datatype-constrained string that locates a service operation
on the Internet by means of a URL.</P></EXPLAIN>
<MODEL><STRING DATATYPE="url"></STRING></MODEL>
</ELEMENTTYPE>

50 <H3>Service Operation Input Document</H3>
<INTRO><P>The input to a service operation is defined by its input document type. That is, the
service operation is invoked when the service operation location receives an input document
whose type corresponds to the document type specified by this element.</P>
<P>Rather than define the expected input and output document types in the market participant

document, this example provides pointers to externally-defined BIDs. This allows reuse of the same BID as the input and/or output document type for multiple operations. In addition, it encourages parallel design and implementation.</P></INTRO>

<ELEMENTTYPE NAME="service.operation.input">

<EXPLAIN><TITLE>Service Operation Input</TITLE>

<SYNOPSIS>Identifies the type of the service operation input document.</SYNOPSIS>

<P>Service location input is a datatype-constrained string that identifies a BID on the Internet by means of a URI.</P>

</EXPLAIN>

<MODEL><STRING DATATYPE="url"></STRING></MODEL>

</ELEMENTTYPE>

<H3>Service Operation Output Document Type</H3>

<INTRO><P>The output of a service operation is defined by its output document type(s). That is, the service operation is expected to emit a document whose type corresponds to the document type specified by this element.</P></INTRO>

<ELEMENTTYPE NAME="service.operation.output">

<EXPLAIN><TITLE>Service Operation Output</TITLE>

<SYNOPSIS>Identifies the type of the service operation output document.</SYNOPSIS>

<P>Service location output is a datatype-constrained string that identifies a BID on the Internet by means of a URI.</P>

</EXPLAIN>

<MODEL><STRING DATATYPE="url"></STRING></MODEL>

</ELEMENTTYPE>

<H3>Service Terms</H3>

<INTRO><P>This is a simple collection of string elements, describing the terms of an agreement.</P></INTRO>

<ELEMENTTYPE NAME="service.terms">

<EXPLAIN><TITLE>Service Terms</TITLE>

<SYNOPSIS>Describes the terms of a given agreement.</SYNOPSIS>

</EXPLAIN>

<MODEL><STRING DATATYPE="string"></STRING></MODEL>

</ELEMENTTYPE>

</DTD>

</SCHEMA>

]

Please alter the paragraph beginning on page 26, line 26, including the computer program listing beginning on page 26, line 29 through page 27, line 4, as follows:

[0065] The service DTD schema may be extended with a service type element in a common business language repository, an example of which is set forth as LISTING 2, in the file named LISTING COMBINED.txt in the accompanying computer program listing appendix.[as follows:

```

5      <!--ELEMENT service.type EMPTY-->
      <!--ATTLIST service.type
          service.type.name (
            catalog.operator
            | commercial.directory.operator
            | eft.services.provider
            | escrower
10         | fulfillment.service
            | insurer
            | manufacturer
            | market.operator
            | order.originator
            | ordering.service
15         | personal.services.provider
            | retailer
            | retail.aggregator
            | schema.resolution.service
            | service.provider
20         | shipment.acceptor
            | shipper
            | van
            | wholesale.aggregator
            ) #REQUIRED
25         %common.attrib;
      >
    ]

```

30 Please alter the paragraph beginning on page 28, line 20, including the computer program listing beginning on page 28, line 23 through page 29, line 9, as follows:

35 [0069] Representative market participant, and service DTDs, created according to the definitions above, are set forth as LISTING 3, in the file named LISTING COMBINED.txt in the accompanying computer program listing appendix. as follows:

```

Market Participant DTD

40 <!--ELEMENT business (party.name+ , address.set) -->
    <!--ATTLIST business business.number CDATA #REQUIRED
        >
    <!--ELEMENT party.name (#PCDATA )>
    <!--ELEMENT city (#PCDATA )>
    <!--ELEMENT internet (#PCDATA )>
45 <!--ELEMENT country (#PCDATA )>
    <!--ELEMENT state (#PCDATA )>

```

5 <!ELEMENT email (#PCDATA)>
 <!ELEMENT address.physical (street , city , state , postcode? , country) >
 <!ELEMENT telephone (#PCDATA)>
 <!ELEMENT person (party.name+ , address.set) >
 <ATTLIST person SSN CDATA #IMPLIED
 >
 <!ELEMENT fax (#PCDATA)>
 <!ELEMENT street (#PCDATA)>
 10 <!ELEMENT address.set (address.physical , telephone* , fax* , email* , internet*) >
 <!ELEMENT postcode (#PCDATA)>
 <!ELEMENT market.participant (business | person) >

 Service DTD
 15 <!ELEMENT service.location (#PCDATA)>
 <!ELEMENT service.terms (#PCDATA)>
 <!ELEMENT service.operation.name (#PCDATA)>
 <!ELEMENT service.operation (service.operation.name , service.operation.location ,
 20 service.operation.input , service.operation.output) >
 <!ELEMENT service (service.name , service.location , service.operation+ , service.terms) >
 <!ELEMENT service.operation.input (#PCDATA)>
 <!ELEMENT service.operation.location (#PCDATA)>
 <!ELEMENT service.name (#PCDATA)>
 <!ELEMENT service.set (service+)>
 25 <!ELEMENT service.operation.output (#PCDATA)>]

Please alter the paragraph beginning on page 29, line 10, including the computer
 program listing beginning on page 29, line 13 through page 31, line 10, as
 follows:

30 **[0070]** One instance of a document produced according to the transact.dtd is set
forth as LISTING 4, in the file named LISTING COMBINED.txt in the
accompanying computer program listing appendix.[follows:

35 <?xml version="1.0"?>
 <!-- rorder.xml Version: 1.0 -->
 <!-- Copyright 1998 Veo Systems, Inc. -->

 40 <!DOCTYPE transaction.description SYSTEM "urn:x-
 veosystems:dtd:cbl:transact:1.0:>
 <transaction.description transaction.type="order">
 <meta>
 <urn?urn:x-veosystems:doc:00023
 </urn>
 45 <thread.id party.assigned.by="reqorg">FRT876
 </thread.id>
 </meta>

5 <issuer.pointer>
 <xll.locator urlink="reqorg.xml">Customer
 Pointer
 </xll.locator>
 </issuer.pointer>
 <counterparty.pointer>
 <xll.locator urlink="compu.xml">Catalog entry owner
 pointer
 </xll.locator>
10 </counterparty.pointer>
 <exchange.description>
 <line.item>
 <product.instance>
 <product.description.pointer>
15 <xll.locator urlink="cthink.xml">Catalogue Entry Pointer
 </xll.locator>
 </product.description.pointer>
 <product.specifics>
 <info.description.set>
20 <info.description>
 <xml.descriptor>
 <doctype>
 <dtd system.id="urn:x-veosystems:dtd:cbl:gprod:1.0"/>
 </doctype>
25 <xml.descriptor.details>
 <xll.xptr.frag>DESCENDANT(ALL,os)STRING("Windows
95")
 </xll.xptr.frag>
 <xll.xptr.frag>DECENDANT(ALL,p.speed)STRING("200")
30 </xll.xptr.frag>
 <xll.xptr.frag>DESCENDANT(ALL,hard.disk.capacity)
 STRING("4")
 </xll.xptr.frag>
 <xll.xptr.frag>DESCENDANT(ALL,d.size)STRING("14.1")
35 </xll.xptr.frag>
 </xml.descriptor.details>
 </xml.descriptor>
 </info.description>
 </info.description.set>
40 </product.specifics>
 <quantity>1
 </quantity>
 </product.instance>
 <shipment.coordinates.set>
45 <shipment.coordinates>
 <shipment.destination>
 <address.set>
 <address.named>SW-1
 </address.named>
50 <address.physical>
 <building.sublocation>208C</building.sublocation>
 <location.in.street>123
 </location.in.street>
 <street>Frontage Rd.

```

5      </street>
      <city>Beltway
      </city>
      <country.subentity.us
country.subentity.us.name="MD"/>
      <postcode>20000
      </postcode>
      </address.physical>
      <telephone>
10     <telephone.number>617-666-2000
      </telephone.number>
      <telephone.extension>1201
      </telephone.extension>
      </telephone>
15     </address.set>
    </shipment.destination>

    <shipment.special>No deliveries after 4 PM</shipment.special>
    </shipment.coordinates>
    </shipment.coordinates.set>
      <payment.set>
      <credit.card
      issuer.name="VISA"
      instrument.number="3787-812345-67893"
25     expiry.date="12/97"
      currency.code="USD"/>
      <amount.group>
      <amount.monetary currency.code="USD">3975
      </amount.monetary>
30     </amount.group>
      </payment.set>
    </line.item>
    </exchange.description>
    </transaction.description>
]
35

```

Please alter the paragraph beginning on page 43, line 5, including the computer program listing beginning on page 43, line 10 days through page 43, line 26, as follows:

[0106] Business interface definitions tell potential trading partners the online services the company offers and which documents to use to invoke those services. Thus, the services are defined in the business interface definition by the documents that they accept and produce. This is illustrated in the [following]fragment of an XML service definition set forth as LISTING 5, in the file named LISTING COMBINED.txt in the accompanying computer

program listing appendix.

```
[    <service>
      <service.name>Order Service</service.name>
      <service.location>www.veosystems.com/order</service.location>
5     <service.op>
      <service.op.name>Submit Order</service.op.name>
      <service.op.inputdoc>www.commerce.net/po.dtd</service.op.inputdoc>
      <service.op.outputdoc>
10         www.veosystems.com/invoice.dtd</service.op.outputdoc>
      </service.op>
      < service.op>
      < service.op.name>Track Order</service.op.name>
      <service.op.inputdoc> www.commerce.net
          /request.track.dtd<service.op.inputdoc>
15     <service.op.outputdoc>
          www.veosystems.com/response.track.dtd<service.op.outputdoc>
      </service.op>
      </service>]
```

Please alter the paragraph beginning on page 47, line 20, including the computer program listing beginning on page 47, line 23 through page 78, line 6, as follows:

[0123] For the example market participant and service DTDs set forth above, the JAVA beans generated by the compiler are set forth (with some redactions for conciseness) as LISTING 6, in the file named LISTING COMBINED.txt in the accompanying computer program listing appendix.[follows:

```
import com.veo.vsp.doclet.meta.Document;
public class AddressPhysical extends Document {
    public static final String DOC_TYPE = "address.physical";
30     String mStreet;
    String mCity;
    public final static int AK = 0;
    public final static int AL = 1;
    public final static int AR = 2;
35     public final static int AZ = 3;
    public final static int CA = 4;
    ...
}
```

```

public final static int WI = 48;
public final static int WV = 49;
public final static int WY = 50;
    int mState;
5      String mPostcode;
      public final static int AD = 51;
      public final static int AE = 52;
      public final static int AF = 53;
      public final static int AG = 54;
10     public final static int AI = 55;
      public final static int AM = 56;
      ...

      int mCountry;
15     public AddressPhysical(){
          super(DOC_TYPE);
          mStreet = new String();
          mCity = new String();
          this.mState = -1;
20     mPostcode = null;
          this.mCountry = -1;
      }
      public AddressPhysical(String doc_type){
          super(doc_type);
25     mStreet = new String();
          mCity = new String();
          this.mState = -1;
          mPostcode = null;
          this.mCountry = -1;
30     }
      static public AddressPhysical initAddressPhysical(String iStreet,String iCity,int
iState,String iPostcode,int iCountry){
          AddressPhysical obj = new AddressPhysical();
          obj.initializeAll(iStreet, iCity, iState, iPostcode, iCountry);
35     return obj;
      }

      public void initializeAll(String iStreet,String iCity,int iState,String iPostcode,int
iCountry){
40     mStreet = iStreet;
          mCity = iCity;
          mState = iState;
          mPostcode = iPostcode;
          mCountry = iCountry;
45     }
      public String getStreet(){
          return mStreet;
      }
      public String getStreetToXML(){
50     if (getStreet() == null) return null;
          char [] c = getStreet().toCharArray();
          StringBuffer sb = new StringBuffer();
          for (int x = 0; x < c.length; x++){
              switch(c[x]){

```

```

        case '>':
            sb.append("&gt;");
            break;
        case '<':
            sb.append("&lt;");
            break;
        case '&':
            sb.append("&amp;");
            break;
        case '"':
            sb.append("&quot;");
            break;
        case '\\':
            sb.append("&quot;");
            break;
        default:
            if (Character.isDefined(c[x]))
                sb.append(c[x]);
    }
    return sb.toString();
}
public void setStreet(String inp){
    this.mStreet = inp;
}
public void streetFromXML(String n){
    setStreet(n);
}
public String getCity(){
    return mCity;
}
public String getCityToXML(){
    if (getCity() == null) return null;
    char [] c = getCity().toCharArray();
    StringBuffer sb = new StringBuffer();
    for (int x = 0; x < c.length; x++){
        switch(c[x]){
            case '>':
                sb.append("&gt;");
                break;
            case '<':
                sb.append("&lt;");
                break;
            case '&':
                sb.append("&amp;");
                break;
            case '"':
                sb.append("&quot;");
                break;
            case '\\':
                sb.append("&quot;");
                break;
            default:
                if (Character.isDefined(c[x]))

```



```

        sb.append(c[x]);
    }
}
return sb.toString();
5    }
    public void setCity(String inp){
        this.mCity = inp;
    }
    public void cityFromXML(String n){
10        setCity(n);
    }
    public int getState(){
        return mState;
    }
15    public String getStateToXML(){
        switch (mState){
            case AK: return "AK";
            case AL: return "AL";
            case AR: return "AR";
20            case AZ: return "AZ";
            ...

        }
        return null;
25    }
    public void setState(int inp){
        this.mState = inp;
    }
    public void stateFromXML(String s){
30        if (s.equals("AK")) mState = AK;
        else if (s.equals("AL"))mState = AL;
        else if (s.equals("AR"))mState = AR;
        else if (s.equals("AZ"))mState = AZ;
        ...
35    }
    public String getPostcode(){
        return mPostcode;
    }
40    public String getPostcodeToXML(){
        if (getPostcode() == null) return null;
        char [] c = getPostcode().toCharArray();
        StringBuffer sb = new StringBuffer();
        for (int x = 0; x < c.length; x++){
45            switch(c[x]){
                case '>':
                    sb.append("&gt;");
                    break;
                case '<':
50                    sb.append("&lt;");
                    break;
                case '&':
                    sb.append("&amp;");
                    break;

```

```

        case "":
            sb.append("&quot;");
            break;
        case "\":
            sb.append("&quot;");
            break;
        default:
            if (Character.isDefined(c[x]))
                sb.append(c[x]);
    }
    }
    return sb.toString();
}
public void setPostcode(String inp){
    this.mPostcode = inp;
}
public void postcodeFromXML(String n){
    setPostcode(n);
}
public int getCountry(){
    return mCountry;
}
public String getCountryToXML(){
    switch (mCountry){
        case AD: return "AD";
        case AE: return "AE";
        case AF: return "AF";
        ...
    }
    return null;
}
public void setCountry(int inp){
    this.mCountry = inp;
}
public void countryFromXML(String s){
    if (s.equals("AD")) mCountry = AD;
    else if (s.equals("AE"))mCountry = AE;
    else if (s.equals("AF"))mCountry = AF;
    else if (s.equals("AG"))mCountry = AG;
    else if (s.equals("AI"))mCountry = AI;
    ..
}
}

```

```

package com.veo.xdk.dev.schema.test.blib;

```

```

import com.veo.vsp.doclet.meta.Document;
public class AddressSet extends Document {
    public static final String DOC_TYPE = "address.set";
    AddressPhysical mAddressPhysical;
    String [] mTelephone;
    String [] mFax;
}

```

```

String [] mEmail;
String [] mInternet;
public AddressSet(){
    super(DOC_TYPE);
    this.mAddressPhysical = new AddressPhysical();
mTelephone = null;
mFax = null;
mEmail = null;
mInternet = null;
}
public AddressSet(String doc_type){
    super(doc_type);
    this.mAddressPhysical = new AddressPhysical();
mTelephone = null;
mFax = null;
mEmail = null;
mInternet = null;
}
static public AddressSet initAddressSet(AddressPhysical iAddressPhysical,String []
iTelephone,String [] iFax,String [] iEmail,String [] iInternet){
    AddressSet obj = new AddressSet();
    obj.initializeAll(iAddressPhysical, iTelephone, iFax, iEmail, iInternet);
    return obj;
}

public void initializeAll(AddressPhysical iAddressPhysical,String [] iTelephone,String
[] iFax,String [] iEmail,String [] iInternet){
    mAddressPhysical = iAddressPhysical;
    mTelephone = iTelephone;
    mFax = iFax;
    mEmail = iEmail;
    mInternet = iInternet;
}
public AddressPhysical getAddressPhysical(){
    return mAddressPhysical;
}
public void setAddressPhysical(AddressPhysical inp){
    this.mAddressPhysical = inp;
}
public String [] getTelephone(){
    return mTelephone;
}
public String getTelephone(int index){
    if (this.mTelephone == null)
        return null;
    if (index >= this.mTelephone.length)
        return null;
    if (index < 0 && -index > this.mTelephone.length)
        return null;
    if (index >= 0) return this.mTelephone[index];
    return this.mTelephone[this.mTelephone.length + index];
}
public String [] getTelephoneToXML(){
    String [] valArr = getTelephone();

```

```

        if (valArr == null) return null;
        String [] nvArr = new String[valArr.length];
        for (int z = 0; z < nvArr.length; z++){
            char [] c = valArr[z].toCharArray();
            StringBuffer st = new StringBuffer();
            StringBuffer sb = new StringBuffer();
            for (int x = 0; x < c.length; x++){
                switch(c[x]){
                    case '>':
                        sb.append("&gt;");
                        break;
                    case '<':
                        sb.append("&lt;");
                        break;
                    case '&':
                        sb.append("&amp;");
                        break;
                    case '"':
                        sb.append("&quot;");
                        break;
                    case "'":
                        sb.append("&apos;");
                        break;
                    default:
                        if (Character.isDefined(c[x]))
                            sb.append(c[x]);
                }
            }
            nvArr[z] = sb.toString();
        }
        return nvArr;
    }

    public void setTelephone(int index, String inp){
        if (this.mTelephone == null) {
            if (index < 0) {
                this.mTelephone = new String[1];
                this.mTelephone[0] = inp;
            } else {
                this.mTelephone = new String[index + 1];
                this.mTelephone[index] = inp;
            }
        } else if (index < 0) {
            String [] newTelephone = new String[this.mTelephone.length + 1];
            java.lang.System.arraycopy((Object)mTelephone, 0,
            (Object)newTelephone, 0, this.mTelephone.length);
            newTelephone[newTelephone.length - 1] = inp;
            mTelephone = newTelephone;
        } else if (index >= this.mTelephone.length){
            String [] newTelephone = new String[index + 1];
            java.lang.System.arraycopy((Object)mTelephone, 0,
            (Object)newTelephone, 0, this.mTelephone.length);
            newTelephone[index] = inp;
            mTelephone = newTelephone;
        } else {

```

```

        this.mTelephone[index] = inp;
    }
}
5   public void setTelephone(String [] inp){
        this.mTelephone = inp;
    }
    public void telephoneFromXML(String n){
        setTelephone(-1, n);
    }
10  public String [] getFax(){
        return mFax;
    }
    public String getFax(int index){
15      if (this.mFax == null)
        return null;
        if (index >= this.mFax.length)
            return null;
        if (index < 0 && -index > this.mFax.length)
            return null;
20      if (index >= 0) return this.mFax[index];
        return this.mFax[this.mFax.length + index];
    }
    public String [] getFaxToXML(){
25      String [] valArr = getFax();
        if (valArr == null) return null;
        String [] nvArr = new String[valArr.length];
        for (int z = 0; z < nvArr.length; z++){
            char [] c = valArr[z].toCharArray();
            StringBuffer st = new StringBuffer();
30          StringBuffer sb = new StringBuffer();
            for (int x = 0; x < c.length; x++){
                switch(c[x]){
                    case '>':
35                     sb.append("&gt;");
                        break;
                    case '<':
                        sb.append("&lt;");
                        break;
                    case '&':
40                     sb.append("&amp;");
                        break;
                    case '"':
                        sb.append("&quot;");
                        break;
45                     case "'":
                        sb.append("&apos;");
                        break;
                    default:
                        if (Character.isDefined(c[x]))
50                         sb.append(c[x]);
                }
            }
            nvArr[z] = sb.toString();
        }
    }
}

```

```

        return nvArr;
    }
    public void setFax(int index, String inp){
        if (this.mFax == null) {
5           if (index < 0) {
                this.mFax = new String[1];
                this.mFax[0] = inp;
            } else {
10             this.mFax = new String[index + 1];
                this.mFax[index] = inp;
            }
        } else if (index < 0) {
                String [] newFax = new String[this.mFax.length + 1];
                java.lang.System.arraycopy((Object)mFax, 0, (Object)newFax, 0,
15             this.mFax.length);
                newFax[newFax.length - 1] = inp;
                mFax = newFax;
            } else if (index >= this.mFax.length){
                String [] newFax = new String[index + 1];
20             java.lang.System.arraycopy((Object)mFax, 0, (Object)newFax, 0,
                this.mFax.length);
                newFax[index] = inp;
                mFax = newFax;
            } else {
25             this.mFax[index] = inp;
            }
        }
    }
    public void setFax(String [] inp){
        this.mFax = inp;
30    }
    public void faxFromXML(String n){
        setFax(-1, n);
    }
    public String [] getEmail(){
35     return mEmail;
    }
    public String getEmail(int index){
        if (this.mEmail == null)
            return null;
40     if (index >= this.mEmail.length)
            return null;
        if (index < 0 && -index > this.mEmail.length)
            return null;
        if (index >= 0) return this.mEmail[index];
45     return this.mEmail[this.mEmail.length + index];
    }
    public String [] getEmailToXML(){
        String [] valArr = getEmail();
        if (valArr == null) return null;
50     String [] nvArr = new String[valArr.length];
        for (int z = 0; z < nvArr.length; z++){
            char [] c = valArr[z].toCharArray();
            StringBuffer st = new StringBuffer();
            StringBuffer sb = new StringBuffer();

```

```

    for (int x = 0; x < c.length; x++){
        switch(c[x]){
            case '>':
                sb.append("&gt;");
                break;
            case '<':
                sb.append("&lt;");
                break;
            case '&':
                sb.append("&amp;");
                break;
            case '"':
                sb.append("&quot;");
                break;
            case "'":
                sb.append("&apos;");
                break;
            default:
                if (Character.isDefined(c[x]))
                    sb.append(c[x]);
        }
    }
    nvArr[z] = sb.toString();
}
return nvArr;
}

public void setEmail(int index, String inp){
    if (this.mEmail == null) {
        if (index < 0) {
            this.mEmail = new String[1];
            this.mEmail[0] = inp;
        } else {
            this.mEmail = new String[index + 1];
            this.mEmail[index] = inp;
        }
    } else if (index < 0) {
        String [] newEmail = new String[this.mEmail.length + 1];
        java.lang.System.arraycopy((Object)mEmail, 0, (Object)newEmail,
0, this.mEmail.length);
        newEmail[newEmail.length - 1] = inp;
        mEmail = newEmail;
    } else if (index >= this.mEmail.length){
        String [] newEmail = new String[index + 1];
        java.lang.System.arraycopy((Object)mEmail, 0, (Object)newEmail,
0, this.mEmail.length);
        newEmail[index] = inp;
        mEmail = newEmail;
    } else {
        this.mEmail[index] = inp;
    }
}

public void setEmail(String [] inp){
    this.mEmail = inp;
}

```

```

public void emailFromXML(String n){
    setEmail(-1, n);
}
5 public String [] getInternet(){
    return mInternet;
}
public String getInternet(int index){
    if (this.mInternet == null)
        return null;
10 if (index >= this.mInternet.length)
    return null;
    if (index < 0 && -index > this.mInternet.length)
        return null;
    if (index >= 0) return this.mInternet[index];
15 return this.mInternet[this.mInternet.length + index];
}
public String [] getInternetToXML(){
    String [] valArr = getInternet();
    if (valArr == null) return null;
20 String [] nvArr = new String[valArr.length];
    for (int z = 0; z < nvArr.length; z++){
        char [] c = valArr[z].toCharArray();
        StringBuffer st = new StringBuffer();
        StringBuffer sb = new StringBuffer();
25 for (int x = 0; x < c.length; x++){
            switch(c[x]){
                case '>':
                    sb.append("&gt;");
                    break;
30 case '<':
                    sb.append("&lt;");
                    break;
                case '&':
                    sb.append("&amp;");
                    break;
35 case '"':
                    sb.append("&quot;");
                    break;
                case "'":
                    sb.append("&apos;");
                    break;
40 case '\n':
                    sb.append("&#10");
                    break;
                default:
                    if (Character.isDefined(c[x]))
                        sb.append(c[x]);
45 }
            }
        nvArr[z] = sb.toString();
    }
    return nvArr;
50 }
public void setInternet(int index, String inp){
    if (this.mInternet == null) {
        if (index < 0) {
            this.mInternet = new String[1];
        }
    }
}

```



```

        this.mInternet[0] = inp;
    } else {
        this.mInternet = new String[index + 1];
        this.mInternet[index] = inp;
5      }
    } else if (index < 0) {
        String [] newInternet = new String[this.mInternet.length + 1];
        java.lang.System.arraycopy((Object)mInternet, 0,
10      (Object)newInternet, 0, this.mInternet.length);
        newInternet[newInternet.length - 1] = inp;
        mInternet = newInternet;
    } else if (index >= this.mInternet.length){
        String [] newInternet = new String[index + 1];
        java.lang.System.arraycopy((Object)mInternet, 0,
15      (Object)newInternet, 0, this.mInternet.length);
        newInternet[index] = inp;
        mInternet = newInternet;
    } else {
        this.mInternet[index] = inp;
20      }
    }
    public void setInternet(String [] inp){
        this.mInternet = inp;
    }
25      public void internetFromXML(String n){
        setInternet(-1, n);
    }
}

30      package com.veo.xdk.dev.schema.test.blib;

import com.veo.vsp.doclet.meta.Document;
public class Business extends Party {
    public static final String DOC_TYPE = "business";
    String aBusinessNumber;
35      public Business(){
        super(DOC_TYPE);
        aBusinessNumber = new String();
    }
    public Business(String doc_type){
        super(doc_type);
        aBusinessNumber = new String();
    }
    static public Business initBusiness(String iBusinessNumber,String []
45      iPartyName,AddressSet iAddressSet){
        Business obj = new Business();
        obj.initializeAll(iBusinessNumber, iPartyName, iAddressSet);
        return obj;
    }
50      public void initializeAll(String iBusinessNumber,String [] iPartyName,AddressSet
iAddressSet){
        aBusinessNumber = iBusinessNumber;
        super.initializeAll(iPartyName, iAddressSet);
    }
}

```

```

    }
    public String getBusinessNumber(){
        return aBusinessNumber;
    }
5    public String getBusinessNumberToXML(){
        if (getBusinessNumber() == null) return null;
        char [] c = getBusinessNumber().toCharArray();
        StringBuffer sb = new StringBuffer();
        for (int x = 0; x < c.length; x++){
10            switch(c[x]){
                case '>':
                    sb.append("&gt;");
                    break;
                case '<':
15                    sb.append("&lt;");
                    break;
                case '&':
                    sb.append("&amp;");
                    break;
20                case '"':
                    sb.append("&quot;");
                    break;
                case '\\':
                    sb.append("&quot;");
25                    break;
                default:
                    if (Character.isDefined(c[x]))
                        sb.append(c[x]);
            }
30        }
        return sb.toString();
    }
    public void setBusinessNumber(String inp){
        this.aBusinessNumber = inp;
35    }
    public void businessNumberFromXML(String n){
        setBusinessNumber(n);
    }
40 }

```

```

import com.véo.vsp.doclet.meta.Document;
public class Party extends Document {
    public static final String DOC_TYPE = "party";
45    String [] mPartyName;
    AddressSet mAddressSet;
    public Party(){
        super(DOC_TYPE);
        mPartyName = new String[0];
50        this.mAddressSet = new AddressSet();
    }
    public Party(String doc_type){
        super(doc_type);
        mPartyName = new String[0];
    }
}

```

```

        this.mAddressSet = new AddressSet();
    }
    static public Party initParty(String [] iPartyName,AddressSet iAddressSet){
        Party obj = new Party();
5         obj.initializeAll(iPartyName, iAddressSet);
        return obj;
    }

    public void initializeAll(String [] iPartyName,AddressSet iAddressSet){
10         mPartyName = iPartyName;
        mAddressSet = iAddressSet;
    }
    public String [] getPartyName(){
        return mPartyName;
15    }
    public String getPartyName(int index){
        if (this.mPartyName == null)
            return null;
        if (index >= this.mPartyName.length)
20            return null;
        if (index < 0 && -index > this.mPartyName.length)
            return null;
        if (index >= 0) return this.mPartyName[index];
        return this.mPartyName[this.mPartyName.length + index];
25    }
    public String [] getPartyNameToXML(){
        String [] valArr = getPartyName();
        if (valArr == null) return null;
        String [] nvArr = new String[valArr.length];
30        for (int z = 0; z < nvArr.length; z++){
            char [] c = valArr[z].toCharArray();
            StringBuffer st = new StringBuffer();
            StringBuffer sb = new StringBuffer();
            for (int x = 0; x < c.length; x++){
35                switch(c[x]){
                    case '>':
                        sb.append("&gt;");
                        break;
                    case '<':
40                        sb.append("&lt;");
                        break;
                    case '&':
                        sb.append("&amp;");
                        break;
                    case '"':
45                        sb.append("&quot;");
                        break;
                    case "'":
                        sb.append("&apos;");
                        break;
                    default:
50                        if (Character.isDefined(c[x]))
                            sb.append(c[x]);
                }
            }
        }
    }

```

```

        }
        nvArr[z] = sb.toString();
    }
    return nvArr;
5    }
    public void setPartyName(int index, String inp){
        if (this.mPartyName == null) {
            if (index < 0) {
                this.mPartyName = new String[1];
10         this.mPartyName[0] = inp;
            } else {
                this.mPartyName = new String[index + 1];
                this.mPartyName[index] = inp;
            }
15         } else if (index < 0) {
            String [] newPartyName = new String[this.mPartyName.length + 1];
            java.lang.System.arraycopy((Object)mPartyName, 0,
            (Object)newPartyName, 0, this.mPartyName.length);
            newPartyName[newPartyName.length - 1] = inp;
20         mPartyName = newPartyName;
        } else if (index >= this.mPartyName.length){
            String [] newPartyName = new String[index + 1];
            java.lang.System.arraycopy((Object)mPartyName, 0,
            (Object)newPartyName, 0, this.mPartyName.length);
25         newPartyName[index] = inp;
            mPartyName = newPartyName;
        } else {
            this.mPartyName[index] = inp;
        }
30     }
    public void setPartyName(String [] inp){
        this.mPartyName = inp;
    }
    public void partyNameFromXML(String n){
35         setPartyName(-1, n);
    }
    public AddressSet getAddressSet(){
        return mAddressSet;
    }
40     public void setAddressSet(AddressSet inp){
        this.mAddressSet = inp;
    }
}

45 package com.veo.xdk.dev.schema.test.blib;

import com.veo.vsp.doclet.meta.Document;
public class Person extends Party {
    public static final String DOC_TYPE = "person";
50     String aSSN;
    public Person(){
        super(DOC_TYPE);
        aSSN = null;
    }
}

```

```

    public Person(String doc_type){
        super(doc_type);
        aSSN = null;
    }
5      static public Person initPerson(String iSSN,String [] iPartyName,AddressSet
iAddressSet){
        Person obj = new Person();
        obj.initializeAll(iSSN, iPartyName, iAddressSet);
        return obj;
10      }

    public void initializeAll(String iSSN,String [] iPartyName,AddressSet iAddressSet){
        aSSN = iSSN;
        super.initializeAll(iPartyName, iAddressSet);
15      }
    public String getSSN(){
        return aSSN;
    }
    public String getSSNToXML(){
20      if (getSSN() == null) return null;
        char [] c = getSSN().toCharArray();
        StringBuffer sb = new StringBuffer();
        for (int x = 0; x < c.length; x++){
            switch(c[x]){
25      case '>':
                sb.append("&gt;");
                break;
            case '<':
                sb.append("&lt;");
                break;
30      case '&':
                sb.append("&amp;");
                break;
            case '"':
                sb.append("&quot;");
                break;
35      case '\\':
                sb.append("&backslash;");
                break;
            default:
                if (Character.isDefined(c[x]))
                    sb.append(c[x]);
40      }
            }
        }
        return sb.toString();
45      }
    }
    public void setSSN(String inp){
        this.aSSN = inp;
    }
50      public void sSNFromXML(String n){
        setSSN(n);
    }
}

```

```

package com.veo.xdk.dev.schema.test.blib;

import com.veo.vsp.doclet.meta.Document;
public class PrototypeService extends Document {
5      public static final String DOC_TYPE = "prototype.service";
      String mServiceName;
      String [] mServiceTerms;
      String [] mServiceLocation;
      ServiceOperation [] mServiceOperation;
10     public PrototypeService(){
          super(DOC_TYPE);
          mServiceName = new String();
          mServiceTerms = new String[0];
          mServiceLocation = new String[0];
15          this.mServiceOperation = new ServiceOperation[0];
      }
      public PrototypeService(String doc_type){
          super(doc_type);
          mServiceName = new String();
20          mServiceTerms = new String[0];
          mServiceLocation = new String[0];
          this.mServiceOperation = new ServiceOperation[0];
      }
      static public PrototypeService initPrototypeService(String iServiceName,String []
25     iServiceTerms,String [] iServiceLocation,ServiceOperation [] iServiceOperation){
          PrototypeService obj = new PrototypeService();
          obj.initializeAll(iServiceName, iServiceTerms, iServiceLocation,
iServiceOperation);
          return obj;
30      }

      public void initializeAll(String iServiceName,String [] iServiceTerms,String []
iServiceLocation,ServiceOperation [] iServiceOperation){
          mServiceName = iServiceName;
35          mServiceTerms = iServiceTerms;
          mServiceLocation = iServiceLocation;
          mServiceOperation = iServiceOperation;
      }
      public String getServiceName(){
40          return mServiceName;
      }
      public String getServiceNameToXML(){
          if (getServiceName() == null) return null;
          char [] c = getServiceName().toCharArray();
45          StringBuffer sb = new StringBuffer();
          for (int x = 0; x < c.length; x++){
              switch(c[x]){
                  case '>':
50                      sb.append("&gt;");
                      break;
                  case '<':
                      sb.append("&lt;");
                      break;
                  case '&':

```

```

        sb.append("&");
        break;
        case "'":
        5         sb.append("&quot;");
            break;
        case "\":
            sb.append("&quot;");
            break;
        10         default:
            if (Character.isDefined(c[x]))
                sb.append(c[x]);
            }
        }
        return sb.toString();
    15     }
    public void setServiceName(String inp){
        this.mServiceName = inp;
    }
    public void serviceNameFromXML(String n){
    20         setServiceName(n);
    }
    public String [] getServiceTerms(){
        return mServiceTerms;
    }
    25     public String getServiceTerms(int index){
        if (this.mServiceTerms == null)
            return null;
        if (index >= this.mServiceTerms.length)
            return null;
    30         if (index < 0 && -index > this.mServiceTerms.length)
            return null;
        if (index >= 0) return this.mServiceTerms[index];
        return this.mServiceTerms[this.mServiceTerms.length + index];
    }
    35     public String [] getServiceTermsToXML(){
        String [] valArr = getServiceTerms();
        if (valArr == null) return null;
        String [] nvArr = new String[valArr.length];
        for (int z = 0; z < nvArr.length; z++){
    40         char [] c = valArr[z].toCharArray();
            StringBuffer st = new StringBuffer();
            StringBuffer sb = new StringBuffer();
            for (int x = 0; x < c.length; x++){
                switch(c[x]){
    45                 case '>':
                    sb.append("&gt;");
                    break;
                    case '<':
                    sb.append("&lt;");
    50                 break;
                    case '&':
                    sb.append("&amp;");
                    break;
                    case "'":

```

```

        sb.append("&quot;");
        break;
    case "":
        sb.append("&quot;");
        break;
    default:
        if (Character.isDefined(c[x]))
            sb.append(c[x]);
        }
    }
    nvArr[z] = sb.toString();
}
return nvArr;
}

15 public void setServiceTerms(int index, String inp){
    if (this.mServiceTerms == null) {
        if (index < 0) {
            this.mServiceTerms = new String[1];
            this.mServiceTerms[0] = inp;
20         } else {
            this.mServiceTerms = new String[index + 1];
            this.mServiceTerms[index] = inp;
        }
    } else if (index < 0) {
25         String [] newServiceTerms = new String[this.mServiceTerms.length
+ 1];
        java.lang.System.arraycopy((Object)mServiceTerms, 0,
        (Object)newServiceTerms, 0, this.mServiceTerms.length);
        newServiceTerms[newServiceTerms.length - 1] = inp;
30         mServiceTerms = newServiceTerms;
    } else if (index >= this.mServiceTerms.length){
        String [] newServiceTerms = new String[index + 1];
        java.lang.System.arraycopy((Object)mServiceTerms, 0,
        (Object)newServiceTerms, 0, this.mServiceTerms.length);
35         newServiceTerms[index] = inp;
        mServiceTerms = newServiceTerms;
    } else {
        this.mServiceTerms[index] = inp;
    }
40 }

    public void setServiceTerms(String [] inp){
        this.mServiceTerms = inp;
    }

    public void serviceTermsFromXML(String n){
45         setServiceTerms(-1, n);
    }

    public String [] getServiceLocation(){
        return mServiceLocation;
    }

50 public String getServiceLocation(int index){
    if (this.mServiceLocation == null)
        return null;
    if (index >= this.mServiceLocation.length)
        return null;

```



```

        if (index < 0 && -index > this.mServiceLocation.length)
            return null;
        if (index >= 0) return this.mServiceLocation[index];
        return this.mServiceLocation[this.mServiceLocation.length + index];
5      }
      public String [] getServiceLocationToXML(){
        String [] valArr = getServiceLocation();
        if (valArr == null) return null;
        String [] nvArr = new String[valArr.length];
10      for (int z = 0; z < nvArr.length; z++){
        char [] c = valArr[z].toCharArray();
        StringBuffer st = new StringBuffer();
        StringBuffer sb = new StringBuffer();
        for (int x = 0; x < c.length; x++){
15      switch(c[x]){
        case '>':
            sb.append("&gt;");
            break;
        case '<':
20      sb.append("&lt;");
            break;
        case '&':
            sb.append("&amp;");
            break;
25      case '"':
            sb.append("&quot;");
            break;
        case "'":
            sb.append("&apos;");
            break;
30      default:
            if (Character.isDefined(c[x]))
                sb.append(c[x]);
            }
        }
        nvArr[z] = sb.toString();
        }
        return nvArr;
    }
40    public void setServiceLocation(int index, String inp){
        if (this.mServiceLocation == null) {
            if (index < 0) {
                this.mServiceLocation = new String[1];
                this.mServiceLocation[0] = inp;
45      } else {
                this.mServiceLocation = new String[index + 1];
                this.mServiceLocation[index] = inp;
            }
        } else if (index < 0) {
50      String [] newServiceLocation = new
String[this.mServiceLocation.length + 1];
                java.lang.System.arraycopy((Object)mServiceLocation, 0,
                (Object)newServiceLocation, 0, this.mServiceLocation.length);
                newServiceLocation[newServiceLocation.length - 1] = inp;

```

```

        mServiceLocation = newServiceLocation;
    } else if (index >= this.mServiceLocation.length){
        String [] newServiceLocation = new String[index + 1];
        java.lang.System.arraycopy((Object)mServiceLocation, 0,
5  (Object)newServiceLocation, 0, this.mServiceLocation.length);
        newServiceLocation[index] = inp;
        mServiceLocation = newServiceLocation;
    } else {
        this.mServiceLocation[index] = inp;
10  }
    }
    public void setServiceLocation(String [] inp){
        this.mServiceLocation = inp;
    }
15  public void serviceLocationFromXML(String n){
        setServiceLocation(-1, n);
    }
    public ServiceOperation [] getServiceOperation(){
        return mServiceOperation;
20  }
    public ServiceOperation getServiceOperation(int index){
        if (this.mServiceOperation == null)
            return null;
        if (index >= this.mServiceOperation.length)
25  return null;
        if (index < 0 && -index > this.mServiceOperation.length)
            return null;
        if (index >= 0) return this.mServiceOperation[index];
        return this.mServiceOperation[this.mServiceOperation.length + index];
30  }
    public void setServiceOperation(int index, ServiceOperation inp){
        if (this.mServiceOperation == null) {
            if (index < 0) {
                this.mServiceOperation = new ServiceOperation[1];
                this.mServiceOperation[0] = inp;
35  } else {
                this.mServiceOperation = new ServiceOperation[index + 1];
                this.mServiceOperation[index] = inp;
            }
        }
        } else if (index < 0) {
40  ServiceOperation [] newServiceOperation = new
ServiceOperation[this.mServiceOperation.length + 1];
        java.lang.System.arraycopy((Object)mServiceOperation, 0,
        (Object)newServiceOperation, 0, this.mServiceOperation.length);
45  newServiceOperation[newServiceOperation.length - 1] = inp;
        mServiceOperation = newServiceOperation;
    } else if (index >= this.mServiceOperation.length){
        ServiceOperation [] newServiceOperation = new
ServiceOperation[index + 1];
50  java.lang.System.arraycopy((Object)mServiceOperation, 0,
        (Object)newServiceOperation, 0, this.mServiceOperation.length);
        newServiceOperation[index] = inp;
        mServiceOperation = newServiceOperation;
    } else {

```

```

        this.mServiceOperation[index] = inp;
    }
}
public void setServiceOperation(ServiceOperation [] inp){
5      this.mServiceOperation = inp;
    }
}

10 package com.veo.xdk.dev.schema.test.blib;

import com.veo.vsp.doclet.meta.Document;
public class Service extends Document {
    public static final String DOC_TYPE = "service";
    String mServiceName;
15    String mServiceLocation;
    ServiceOperation [] mServiceOperation;
    String mServiceTerms;
    public Service(){
        super(DOC_TYPE);
20    mServiceName = new String();
    mServiceLocation = new String();
        this.mServiceOperation = new ServiceOperation[0];
    mServiceTerms = new String();
    }
25    public Service(String doc_type){
        super(doc_type);
    mServiceName = new String();
    mServiceLocation = new String();
        this.mServiceOperation = new ServiceOperation[0];
30    mServiceTerms = new String();
    }
    static public Service initService(String iServiceName,String
iServiceLocation,ServiceOperation [] iServiceOperation,String iServiceTerms){
        Service obj = new Service();
35    obj.initializeAll(iServiceName, iServiceLocation, iServiceOperation,
iServiceTerms);
        return obj;
    }

40    public void initializeAll(String iServiceName,String
iServiceLocation,ServiceOperation [] iServiceOperation,String iServiceTerms){
        mServiceName = iServiceName;
        mServiceLocation = iServiceLocation;
        mServiceOperation = iServiceOperation;
45    mServiceTerms = iServiceTerms;
    }
    public String getServiceName(){
        return mServiceName;
    }
50    public String getServiceNameToXML(){
        if (getServiceName() == null) return null;
        char [] c = getServiceName().toCharArray();
        StringBuffer sb = new StringBuffer();
        for (int x = 0; x < c.length; x++){

```

```

switch(c[x]){
case '>':
    sb.append("&gt;");
    break;
5 case '<':
    sb.append("&lt;");
    break;
case '&':
    sb.append("&amp;");
10 break;
case '"':
    sb.append("&quot;");
    break;
15 case '\\':
    sb.append("&quot;");
    break;
default:
    if (Character.isDefined(c[x]))
        sb.append(c[x]);
20 }
}
return sb.toString();
}
25 public void setServiceName(String inp){
    this.mServiceName = inp;
}
public void serviceNameFromXML(String n){
    setServiceName(n);
}
30 public String getServiceLocation(){
    return mServiceLocation;
}
public String getServiceLocationToXML(){
    if (getServiceLocation() == null) return null;
35 char [] c = getServiceLocation().toCharArray();
    StringBuffer sb = new StringBuffer();
    for (int x = 0; x < c.length; x++){
        switch(c[x]){
            case '>':
40 sb.append("&gt;");
                break;
            case '<':
                sb.append("&lt;");
                break;
45 case '&':
                sb.append("&amp;");
                break;
            case '"':
                sb.append("&quot;");
50 break;
            case '\\':
                sb.append("&quot;");
                break;
            default:

```

```

        if (Character.isDefined(c[x]))
            sb.append(c[x]);
        }
    }
    return sb.toString();
}

public void setServiceLocation(String inp){
    this.mServiceLocation = inp;
}

public void serviceLocationFromXML(String n){
    setServiceLocation(n);
}

public ServiceOperation [] getServiceOperation(){
    return mServiceOperation;
}

public ServiceOperation getServiceOperation(int index){
    if (this.mServiceOperation == null)
        return null;
    if (index >= this.mServiceOperation.length)
        return null;
    if (index < 0 && -index > this.mServiceOperation.length)
        return null;
    if (index >= 0) return this.mServiceOperation[index];
    return this.mServiceOperation[this.mServiceOperation.length + index];
}

public void setServiceOperation(int index, ServiceOperation inp){
    if (this.mServiceOperation == null) {
        if (index < 0) {
            this.mServiceOperation = new ServiceOperation[1];
            this.mServiceOperation[0] = inp;
        } else {
            this.mServiceOperation = new ServiceOperation[index + 1];
            this.mServiceOperation[index] = inp;
        }
    } else if (index < 0) {
        ServiceOperation [] newServiceOperation = new
ServiceOperation[this.mServiceOperation.length + 1];
        java.lang.System.arraycopy((Object)mServiceOperation, 0,
(Object)newServiceOperation, 0, this.mServiceOperation.length);
        newServiceOperation[newServiceOperation.length - 1] = inp;
        mServiceOperation = newServiceOperation;
    } else if (index >= this.mServiceOperation.length){
        ServiceOperation [] newServiceOperation = new
ServiceOperation[index + 1];
        java.lang.System.arraycopy((Object)mServiceOperation, 0,
(Object)newServiceOperation, 0, this.mServiceOperation.length);
        newServiceOperation[index] = inp;
        mServiceOperation = newServiceOperation;
    } else {
        this.mServiceOperation[index] = inp;
    }
}

public void setServiceOperation(ServiceOperation [] inp){
    this.mServiceOperation = inp;
}

```

```

    }
    public String getServiceTerms(){
        return mServiceTerms;
    }
5    public String getServiceTermsToXML(){
        if (getServiceTerms() == null) return null;
        char [] c = getServiceTerms().toCharArray();
        StringBuffer sb = new StringBuffer();
        for (int x = 0; x < c.length; x++){
10         switch(c[x]){
            case '>':
                sb.append("&gt;");
                break;
            case '<':
15         sb.append("&lt;");
                break;
            case '&':
                sb.append("&amp;");
                break;
20         case '"':
                sb.append("&quot;");
                break;
            case '\\':
                sb.append("&quot;");
25         break;
            default:
                if (Character.isDefined(c[x]))
                    sb.append(c[x]);
        }
30     }
        return sb.toString();
    }
    public void setServiceTerms(String inp){
        this.mServiceTerms = inp;
35     }
    public void serviceTermsFromXML(String n){
        setServiceTerms(n);
    }
40 }

package com.veo.xdk.dev.schema.test.blib;

import com.veo.vsp.doclet.meta.Document;
45 public class ServiceOperation extends Document {
    public static final String DOC_TYPE = "service.operation";
    String mServiceOperationName;
    String mServiceOperationLocation;
    String mServiceOperationInput;
    String mServiceOperationOutput;
50 public ServiceOperation(){
    super(DOC_TYPE);
    mServiceOperationName = new String();
    mServiceOperationLocation = new String();
    mServiceOperationInput = new String();

```

```

        mServiceOperationOutput = new String();
    }
    public ServiceOperation(String doc_type){
        super(doc_type);
5       mServiceOperationName = new String();
        mServiceOperationLocation = new String();
        mServiceOperationInput = new String();
        mServiceOperationOutput = new String();
    }
10      static public ServiceOperation initServiceOperation(String
iServiceOperationName,String iServiceOperationLocation,String iServiceOperationInput,String
iServiceOperationOutput){
        ServiceOperation obj = new ServiceOperation();
        obj.initializeAll(iServiceOperationName, iServiceOperationLocation,
15      iServiceOperationInput, iServiceOperationOutput);
        return obj;
    }

    public void initializeAll(String iServiceOperationName,String
20      iServiceOperationLocation,String iServiceOperationInput,String iServiceOperationOutput){
        mServiceOperationName = iServiceOperationName;
        mServiceOperationLocation = iServiceOperationLocation;
        mServiceOperationInput = iServiceOperationInput;
        mServiceOperationOutput = iServiceOperationOutput;
25      }
    public String getServiceOperationName(){
        return mServiceOperationName;
    }
    public String getServiceOperationNameToXML(){
30      if (getServiceOperationName() == null) return null;
        char [] c = getServiceOperationName().toCharArray();
        StringBuffer sb = new StringBuffer();
        for (int x = 0; x < c.length; x++){
            switch(c[x]){
35      case '>':
                sb.append("&gt;");
                break;
            case '<':
                sb.append("&lt;");
40      case '&':
                sb.append("&amp;");
                break;
            case '"':
                sb.append("&quot;");
45      case '\':
                sb.append("&quot;");
                break;
            default:
                if (Character.isDefined(c[x]))
                    sb.append(c[x]);
50      }
        }
    }
}

```

```

        return sb.toString();
    }
    public void setServiceOperationName(String inp){
        this.mServiceOperationName = inp;
5    }
    public void serviceOperationNameFromXML(String n){
        setServiceOperationName(n);
    }
    public String getServiceOperationLocation(){
10    return mServiceOperationLocation;
    }
    public String getServiceOperationLocationToXML(){
        if (getServiceOperationLocation() == null) return null;
        char [] c = getServiceOperationLocation().toCharArray();
15    StringBuffer sb = new StringBuffer();
        for (int x = 0; x < c.length; x++){
            switch(c[x]){
                case '>':
20                sb.append("&gt;");
                    break;
                case '<':
                    sb.append("&lt;");
                    break;
                case '&':
25                sb.append("&amp;");
                    break;
                case '"':
                    sb.append("&quot;");
                    break;
30                case '\\':
                    sb.append("&backslash;");
                    break;
                default:
                    if (Character.isDefined(c[x]))
35                    sb.append(c[x]);
            }
        }
        return sb.toString();
    }
40    public void setServiceOperationLocation(String inp){
        this.mServiceOperationLocation = inp;
    }
    public void serviceOperationLocationFromXML(String n){
        setServiceOperationLocation(n);
45    }
    public String getServiceOperationInput(){
        return mServiceOperationInput;
    }
    public String getServiceOperationInputToXML(){
50    if (getServiceOperationInput() == null) return null;
        char [] c = getServiceOperationInput().toCharArray();
        StringBuffer sb = new StringBuffer();
        for (int x = 0; x < c.length; x++){
            switch(c[x]){

```



```

        case '>':
            sb.append("&gt;");
            break;
        case '<':
5           sb.append("&lt;");
            break;
        case '&':
            sb.append("&amp;");
            break;
10         case '"':
            sb.append("&quot;");
            break;
        case '\\':
15         sb.append("&backslash;");
            break;
        default:
            if (Character.isDefined(c[x]))
                sb.append(c[x]);
20     }
    }
    return sb.toString();
}

public void setServiceOperationInput(String inp){
    this.mServiceOperationInput = inp;
25 }

public void serviceOperationInputFromXML(String n){
    setServiceOperationInput(n);
}

public String getServiceOperationOutput(){
30     return mServiceOperationOutput;
}

public String getServiceOperationOutputToXML(){
    if (getServiceOperationOutput() == null) return null;
    char [] c = getServiceOperationOutput().toCharArray();
35     StringBuffer sb = new StringBuffer();
    for (int x = 0; x < c.length; x++){
        switch(c[x]){
            case '>':
40             sb.append("&gt;");
                break;
            case '<':
                sb.append("&lt;");
                break;
            case '&':
45             sb.append("&amp;");
                break;
            case '"':
                sb.append("&quot;");
                break;
50             case '\\':
                sb.append("&backslash;");
                break;
            default:
                if (Character.isDefined(c[x]))

```

```

        sb.append(c[x]);
    }
    }
    return sb.toString();
5    }
    public void setServiceOperationOutput(String inp){
        this.mServiceOperationOutput = inp;
    }
    public void serviceOperationOutputFromXML(String n){
10        setServiceOperationOutput(n);
    }
}

package com.veo.xdk.dev.schema.test.blib;

15
import com.veo.vsp.doclet.meta.Document;
public class ServiceSet extends Document {
    public static final String DOC_TYPE = "service.set";
    Service [] mService;
20    public ServiceSet(){
        super(DOC_TYPE);
        this.mService = new Service[0];
    }
    public ServiceSet(String doc_type){
25        super(doc_type);
        this.mService = new Service[0];
    }
    static public ServiceSet initServiceSet(Service [] iService){
        ServiceSet obj = new ServiceSet();
        obj.initializeAll(iService);
30        return obj;
    }

    public void initializeAll(Service [] iService){
35        mService = iService;
    }
    public Service [] getService(){
        return mService;
    }
40    public Service getService(int index){
        if (this.mService == null)
            return null;
        if (index >= this.mService.length)
            return null;
45        if (index < 0 && -index > this.mService.length)
            return null;
        if (index >= 0) return this.mService[index];
        return this.mService[this.mService.length + index];
    }
50    public void setService(int index, Service inp){
        if (this.mService == null) {
            if (index < 0) {
                this.mService = new Service[1];
                this.mService[0] = inp;
            }
        }
    }
}

```

```

        } else {
            this.mService = new Service[index + 1];
            this.mService[index] = inp;
        }
5      } else if (index < 0) {
            Service [] newService = new Service[this.mService.length + 1];
            java.lang.System.arraycopy((Object)mService, 0,
(Object)newService, 0, this.mService.length);
            newService[newService.length - 1] = inp;
10         mService = newService;
        } else if (index >= this.mService.length){
            Service [] newService = new Service[index + 1];
            java.lang.System.arraycopy((Object)mService, 0,
(Object)newService, 0, this.mService.length);
15         newService[index] = inp;
            mService = newService;
        } else {
            this.mService[index] = inp;
        }
20     }
    public void setService(Service [] inp){
        this.mService = inp;
    }
}
25

```

Please alter the paragraph beginning on page 78, line 8, including the computer program listing beginning on page 78, line 11 through page 84, line 22, as follows:

30 **[0124]** In addition to the JAVA beans set forth above, transformation code is produced for translating from JAVA to XML and XML to JAVA as set forth as LISTING 7, in the file named LISTING COMBINED.txt in the accompanying computer program listing appendix.[below:

```

Java to XML
35
<!DOCTYPE tree SYSTEM "tree.dtd">
<tree source = "null" pass-through = "false">
<before>
<vardef name = "attribute.def">
40   <element source = "ATTRIBUTE" class = "NAME" type = "5" position = "-2">
       <parse>
         <data class = "java.lang.String" position = "-2"/>
       </parse>
     </element>
45   </vardef>
   <vardef name = "pcdata.def">

```

```

5      <element source = "PCDATA" class = "NAME" type = "4" position = "-2">
        <parse>
          <data class = "999" type = "6" position = "-2"/>
        </parse>
      </element>
    </vardef>
    <vardef name = "content.def">
      <element source = "PCDATA">
        <parse>
10       <data class = "999" type = "6" position = "-2"/>
        </parse>
      </element>
    </vardef>
    <vardef name = "ServiceSet.var">
15    <element source = "com.veo.xdk.dev.schema.test.blib.ServiceSet" class = "service.set" type =
      "4" position = "-2">
      <parse>
        <callvar name = "Service.var"/>
      </parse>
    </element>
    </vardef>
    <vardef name = "PrototypeService.var">
      <element source = "com.veo.xdk.dev.schema.test.blib.PrototypeService" class =
25    "prototype.service" type = "4" position = "-2">
      <parse>
        <callvar name = "pcdata.def" parms = "setSource ServiceNameToXML setGenerator
          service.name"/>
        <callvar name = "pcdata.def" parms = "setSource ServiceTermsToXML setGenerator
          service.terms"/>
30      <callvar name = "pcdata.def" parms = "setSource ServiceLocationToXML setGenerator
          service.location"/>
        <callvar name = "ServiceOperation.var"/>
      </parse>
    </element>
    </vardef>
35    <vardef name = "Service.var">
      <element source = "com.veo.xdk.dev.schema.test.blib.Service" class = "service" type = "8"
        position = "0">
        <parse>
40      <callvar name = "pcdata.def" parms = "setSource ServiceNameToXML setGenerator
          service.name"/>
        <callvar name = "pcdata.def" parms = "setSource ServiceLocationToXML setGenerator
          service.location"/>
        <callvar name = "ServiceOperation.var"/>
45      <callvar name = "pcdata.def" parms = "setSource ServiceTermsToXML setGenerator
          service.terms"/>
        </parse>
      </element>
    </vardef>
50    <vardef name = "ServiceOperation.var">
      <element source = "com.veo.xdk.dev.schema.test.blib.ServiceOperation" class =
        "service.operation" type = "4" position = "-2">
        <parse>
          <callvar name = "pcdata.def" parms = "setSource ServiceOperationNameToXML

```

```

setGenerator service.operation.name"/>
    <callvar name = "pcdata.def" parms = "setSource ServiceOperationLocationToXML
setGenerator service.operation.location"/>
    <callvar name = "pcdata.def" parms = "setSource ServiceOperationInputToXML
5 setGenerator service.operation.input"/>
    <callvar name = "pcdata.def" parms = "setSource ServiceOperationOutputToXML
setGenerator service.operation.output"/>
    </parse>
    </element>
10 </vardef>
    </before>
    <parse>
    <callvar name = "ServiceSet.var"/>
    <callvar name = "PrototypeService.var"/>
15 <callvar name = "Service.var"/>
    <callvar name = "ServiceOperation.var"/>
    </parse>
    </tree>

20 XML to Java

    <!DOCTYPE tree SYSTEM "tree.dtd">
    <tree source = "null" pass-through = "false">
    <before>
25 <vardef name = "business.var">
    <element source = "business"
        class = "com.veo.xdk.dev.schema.test.blib.Business"
        type = "7" setter = "setBusiness">
    <before>
30 <onattribute name = "business.number">
    <actions>
        <callmeth name = "businessNumberFromXML">
        <parms>
            <getattr name = "business.number"/>
35 </parms>
        </callmeth>
    </actions>
    </onattribute>
    </before>
40 <parse>
    <callvar name = "party.name.var" parms = "setPosition -1"/>
    <callvar name = "address.set.var"/>
    </parse>
    </element>
45 </vardef>
    <vardef name = "party.name.var">
    <element source = "party.name" setter = "partyNameFromXML" position = "-1" class =
"java.lang.String">
    <parse>
50 <data class = "java.lang.String" position = "0"/>
    </parse>
    </element>
    </vardef>
    <vardef name = "city.var">

```

```

5      <element source = "city" setter = "cityFromXML" position = "-1" class = "java.lang.String">
        <parse>
          <data class = "java.lang.String" position = "0"/>
        </parse>
      </element>
    </vardef>
    <vardef name = "internet.var">
      <element source = "internet" setter = "internetFromXML" position = "-1" class =
10      "java.lang.String">
        <parse>
          <data class = "java.lang.String" position = "0"/>
        </parse>
      </element>
    </vardef>
    <vardef name = "country.var">
15      <element source = "country" setter = "countryFromXML" position = "-1" class =
      "java.lang.String">
        <parse>
          <data class = "java.lang.String" position = "0"/>
20      </parse>
        </element>
      </vardef>
    <vardef name = "state.var">
      <element source = "state" setter = "stateFromXML" position = "-1" class = "java.lang.String">
25      <parse>
        <data class = "java.lang.String" position = "0"/>
      </parse>
    </element>
  </vardef>
  <vardef name = "email.var">
30      <element source = "email" setter = "emailFromXML" position = "-1" class =
      "java.lang.String">
        <parse>
          <data class = "java.lang.String" position = "0"/>
35      </parse>
        </element>
      </vardef>
    <vardef name = "address.physical.var">
      <element source = "address.physical"
40      class = "com.veo.xdk.dev.schema.test.blib.AddressPhysical"
      type = "7" setter = "setAddressPhysical">
        <before>
        </before>
        <parse>
45      <callvar name = "street.var" parms = "setPosition -1"/>
      <callvar name = "city.var" parms = "setPosition -1"/>
      <callvar name = "state.var" parms = "setPosition -1"/>
      <callvar name = "postcode.var" parms = "setPosition -1"/>
      <callvar name = "country.var" parms = "setPosition -1"/>
50      </parse>
    </element>
  </vardef>
  <vardef name = "telephone.var">
    <element source = "telephone" setter = "telephoneFromXML" position = "-1" class =

```

```

"java.lang.String">
  <parse>
    <data class = "java.lang.String" position = "0"/>
  </parse>
5  </element>
</vardef>
<vardef name = "person.var">
  <element source = "person"
    class = "com.veo.xdk.dev.schema.test.blib.Person"
10  type = "7" setter = "setPerson">
    <before>
      <onattribute name = "SSN">
        <actions>
          <callmeth name = "sSNFromXML">
15  <parms>
            <getattr name = "SSN"/>
          </parms>
        </callmeth>
      </actions>
    </onattribute>
  </before>
  <parse>
    <callvar name = "party.name.var" parms = "setPosition -1"/>
    <callvar name = "address.set.var"/>
25  </parse>
  </element>
</vardef>
<vardef name = "fax.var">
  <element source = "fax" setter = "faxFromXML" position = "-1" class = "java.lang.String">
30  <parse>
    <data class = "java.lang.String" position = "0"/>
  </parse>
  </element>
</vardef>
35  <vardef name = "street.var">
    <element source = "street" setter = "streetFromXML" position = "-1" class =
      "java.lang.String">
      <parse>
        <data class = "java.lang.String" position = "0"/>
40  </parse>
      </element>
    </vardef>
    <vardef name = "address.set.var">
      <element source = "address.set"
45  class = "com.veo.xdk.dev.schema.test.blib.AddressSet"
        type = "7" setter = "setAddressSet">
        <before>
        </before>
        <parse>
50  <callvar name = "address.physical.var"/>
        <callvar name = "telephone.var" parms = "setPosition -1"/>
        <callvar name = "fax.var" parms = "setPosition -1"/>
        <callvar name = "email.var" parms = "setPosition -1"/>
        <callvar name = "internet.var" parms = "setPosition -1"/>

```

```

    </parse>
  </element>
</vardef>
5  <vardef name = "postcode.var">
    <element source = "postcode" setter = "postcodeFromXML" position = "-1" class =
      "java.lang.String">
      <parse>
        <data class = "java.lang.String" position = "0"/>
      </parse>
10  </element>
    </vardef>
    <vardef name = "market.participant.var">
      <element source = "market.participant"
        class = "com.veo.xdk.dev.schema.test.blib.MarketParticipant"
15      type = "7" position = "0">
        <before>
        </before>
        <parse>
          <callvar name = "business.var"/>
20      <callvar name = "person.var"/>
        </parse>
        </element>
    </vardef>
    </before>
25  <parse>
    <callvar name = "business.var"/>
    <callvar name = "party.name.var"/>
    <callvar name = "city.var"/>
    <callvar name = "internet.var"/>
30  <callvar name = "country.var"/>
    <callvar name = "state.var"/>
    <callvar name = "email.var"/>
    <callvar name = "address.physical.var"/>
    <callvar name = "telephone.var"/>
35  <callvar name = "person.var"/>
    <callvar name = "fax.var"/>
    <callvar name = "street.var"/>
    <callvar name = "address.set.var"/>
    <callvar name = "postcode.var"/>
40  <callvar name = "market.participant.var"/>
    </parse>
  </tree>

```

Makefiles:

```

45  #
    # this makefile was generated by bic version 0.0. 05/02/1998
    #
50  #
    #

```



```
# get the package name from the package argument passed to SchemaGen
PACKAGE_NAME = com/veo/xdk/dev/schema/test/blib
```

```
5  JAVA_SOURCES += \
    MarketParticipant.java \
    Business.java \
    Person.java \
    Party.java \
    AddressPhysical.java \
10  AddressSet.java \
```

```
MAKEFILE_MASTER_DIR = xxx
include $(MAKEFILE_MASTER_DIR)/Makefile.master
```

```
15  all:: $(JAVA_CLASSES)
```

```
#
# this makefile was generated by bic version 0.0. 05/02/1998
#
20  #
#
#
```

```
# get the package name from the package argument passed to SchemaGen
PACKAGE_NAME = com/veo/xdk/dev/schema/test/blib
```

```
25  JAVA_SOURCES += \
    ServiceSet.java \
    PrototypeService.java \
30  Service.java \
    ServiceOperation.java \
```

```
MAKEFILE_MASTER_DIR = xxx
include $(MAKEFILE_MASTER_DIR)/Makefile.master
```

```
35  all:: $(JAVA_CLASSES) ]
```

Please alter the paragraph beginning on page 84, line 24, including the computer program listing beginning on page 84, line 28 through page 85, line 29, as follows:

[0125] Finally, the XML document instances generated at run time according to the model above for one example is set forth as LISTING 8, in the file named LISTING COMBINED.txt in the accompanying computer program listing appendix.[follows:

<!DOCTYPE market.participant SYSTEM "market.participant.dtd" >
 <market.participant>
 5 <business business.number="1234567890" >
 <party.name>IBM</party.name>
 <address.set>
 10 <address.physical>
 <street>1 IBM Way</street>
 <city>Palo Alto</city>
 <state>CA</state>
 <postcode>94304</postcode>
 15 <country>USA</country>
 </address.physical>
 <telephone>123 456-7890</telephone>
 <fax>123 456 0987</fax>
 20 <email>ibmec@ibm.com</email>
 </address.set>
 </business>
 25 </market.participant>
 <!DOCTYPE service SYSTEM "service.dtd" >
 <service.set>
 <service>
 30 <service.name>Order Service</service.name>
 <service.location>www.ibm.com/order</service.location>
 <service.operation>
 <service.operation.name>Submit Order</service.operation.name>
 35 <service.operation.location>www.ibm.com/order/submit</service.location>
 <service.operation.input>urn:x-ibm:services:order:operations:po.dtd</service.operation.input>
 <service.operation.output>urn:x-
 ibm:services:order:operations:poack.dtd</service.operation.output>
 </service.operation>
 40 <service.operation>
 <service.operation.name>Track Order</service.operation.name>
 <service.operation.location>www.ibm.com/order/track</service.location>
 <service.operation.input>urn:x-
 45 ibm:services:order:operations:track.irequest.dtd</service.operation.input>
 <service.operation.output>urn:x-
 ibm:services:order:operations:track.iresponse.dtd</service.operation.output>
 </service.operation>
 50 </service>
 </service.set>

]